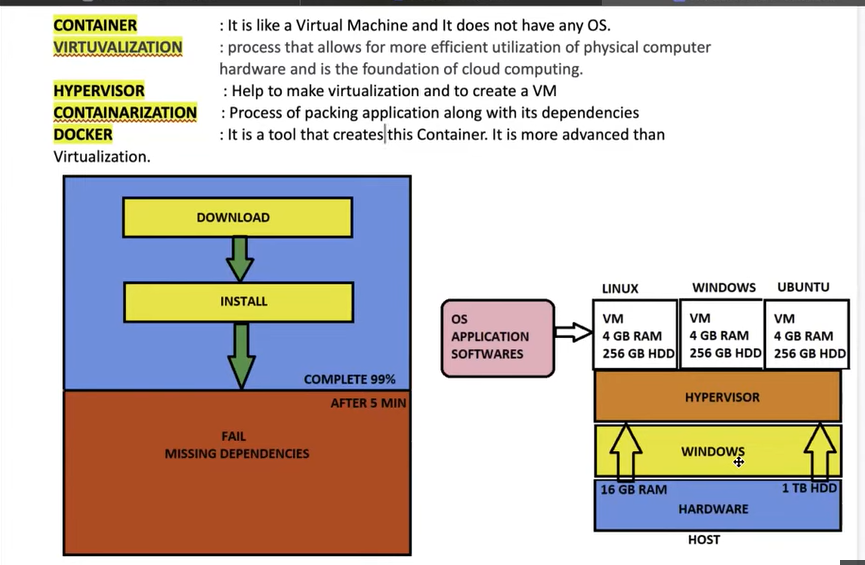
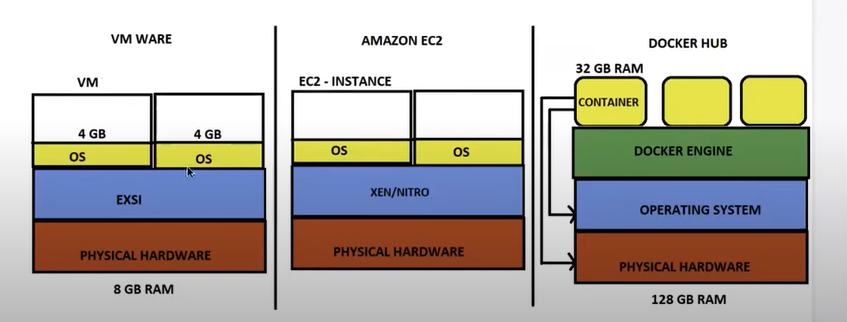
Docker





**Docker:-**

. It is an open source centralized platform designed to create, deploy and run applications.

. Docker is written on GO language.( go:- google official)

. Docker uses container on host O.S to run applications.it allows applications to use same Linux kernel as a system on the host computer, rather than creating a whole virtual O.S.

. We can install Docker on any O.S but Docker engine runs natively on Linux distribution.

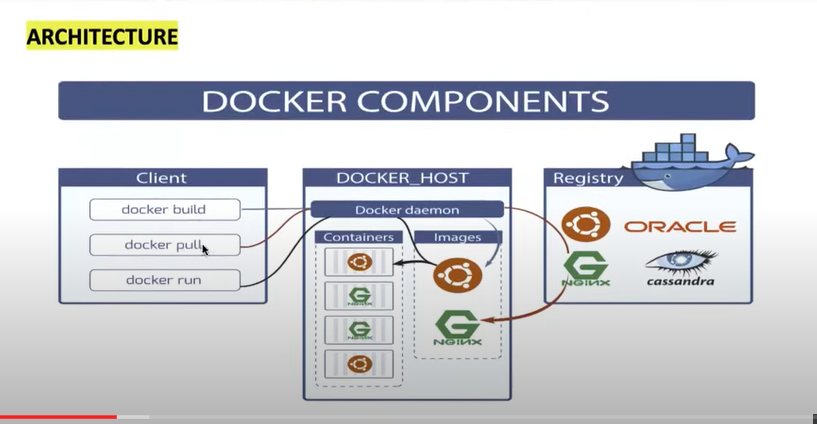
. Docker performs O.S level virtualization also known as containerization.

. Before Docker many users face problems that a particular code is running in the developers system but not in the user system.

. It was initially release in March 2013, and developed by Solomon hykes and Sebastian pahl.

. Docker is a set of platform-as-a-service that use O.S level virtualization, where as VM vare uses hardware level virtualization.

. Container contains O.S files but it’s negligible in size compared to original files of that O.S.



**Docker client:-** is the primary way that many Docker users interact with Docker. When you use commands such as Docker run, the clients sends these commands to Docker daemon, which carries them out. The Docker command uses the Docker API.

**Docker Host:-** Docker host is the machine where you installed the Docker engine.

**Docker Deamon:-** Docker daemon runs on the host operating system. It is responsible for running containers to manage Docker services. Docker daemon communicates with other daemon it offers various Docker objects such as images, containers, networking, and storage.

Container topic:- small reminder

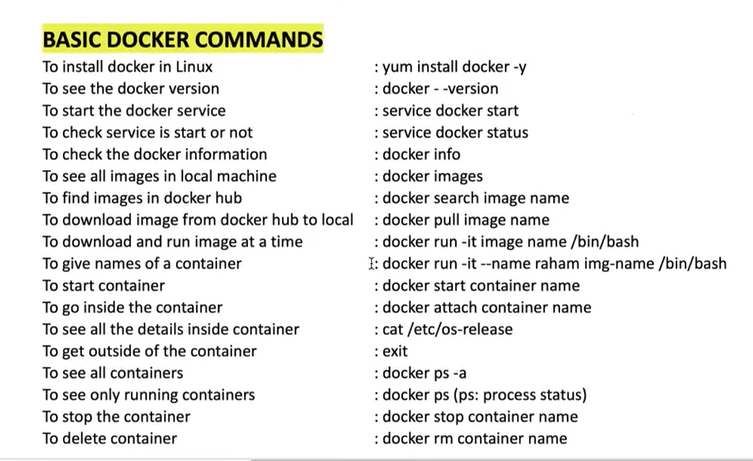
. how to exit without stopping container🡺 ctrlp & ctrlq

. how to enter container . docker exec –it <contaierid> /bin/bash

.how to check complete history of container🡺 docker logs <containerid>

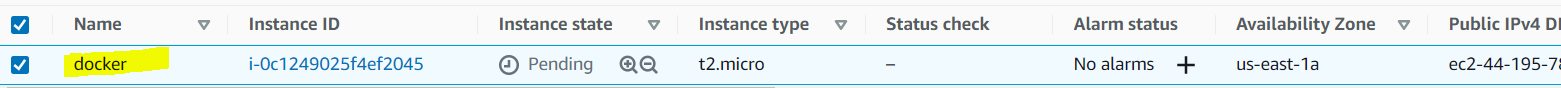
.how to change name and port in container

. docker run –d –p<Newport>:<oldport> --name <newname> <osname>



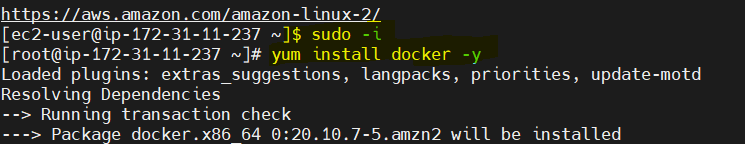
Installing docker:-

. Launch one instance



. open mobaxterm🡺 install docker

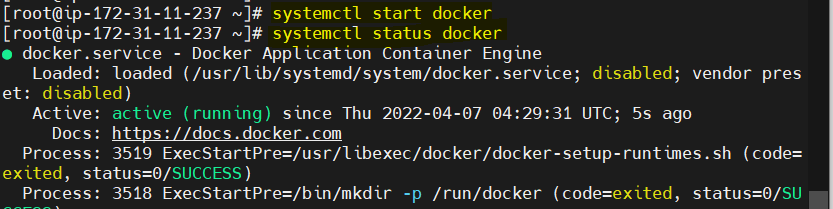
. yum install docker –y



. start docker

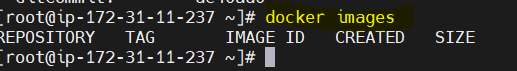
. systemctl start docker

. systemctl status docker



. to see docker images on client (client means our local machine)

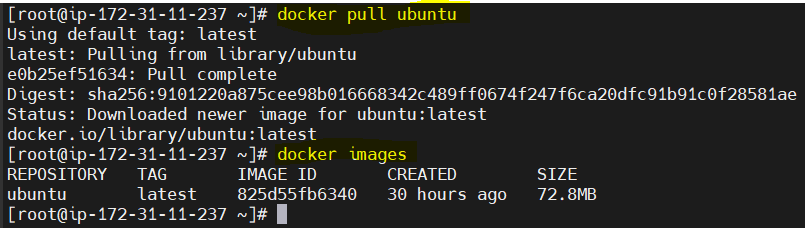
. docker images



. how u clone image

. docker pull <image name>

Ex:- docker pull ubuntu

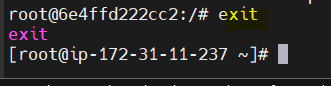


. how to create container through image

. docker run –it Ubuntu /bin/bash (it:- iteractive mode)



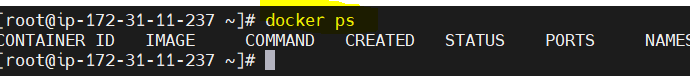
. how to exit container



. when you exit container, container service stoped(inactive).

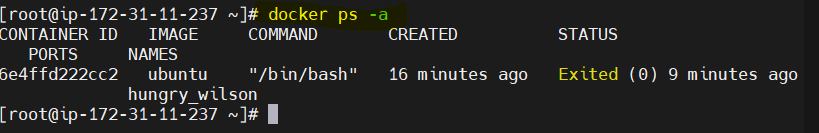
. how to see containers list

. docker ps

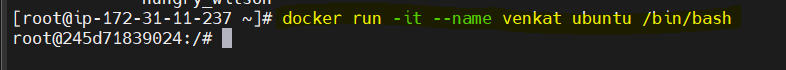


. to show all the containers list active or inactive list.

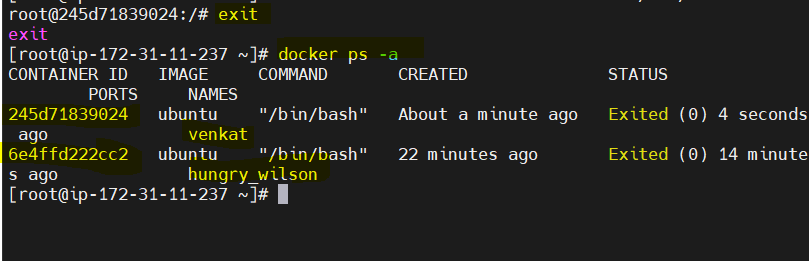
. docker ps -a



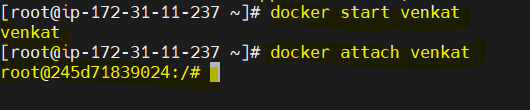
. how to give name to container



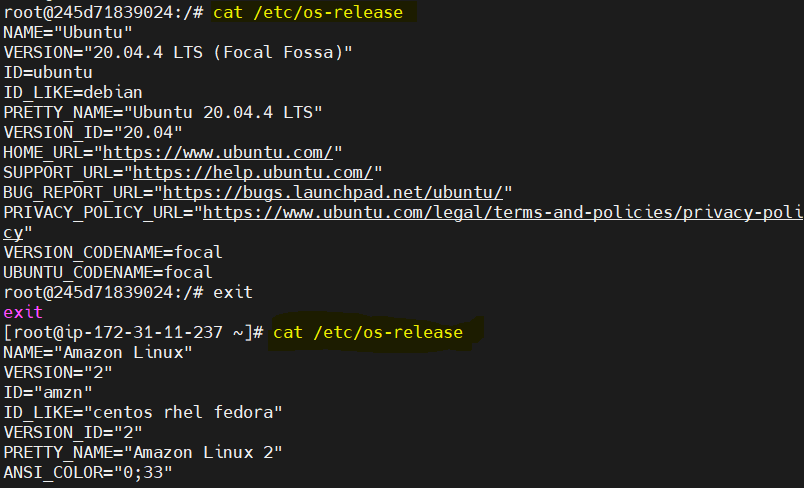
. when u exit container🡺 see docker ps –a🡺 to list the containers 🡺 but u run only one image(ubuntu) 🡺 but u exit on containers🡺 container id is change



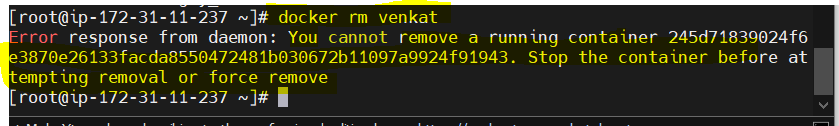
. how to start container

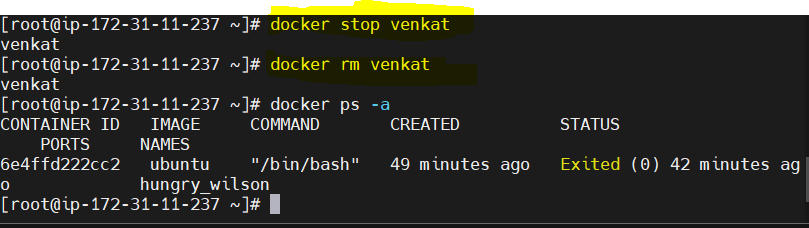


. to see a operating system in container



. how to remove container (you can not delete running container)





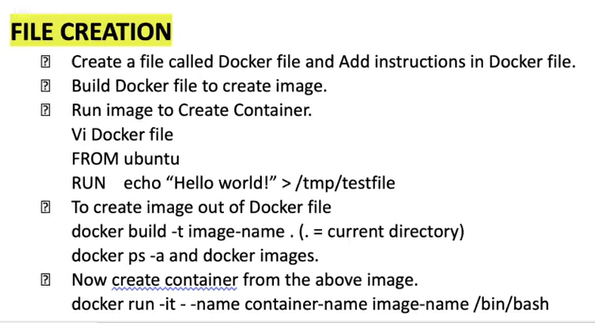
**Docker file:-**

. It is basically a text file which contains some set of instructions.

. Automation of Docker image creation.

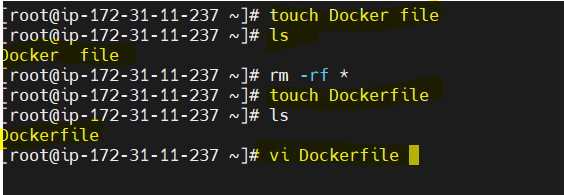
. Always D is capital letter on Dockerfile.( there is no separation(space) on Dockerfile)

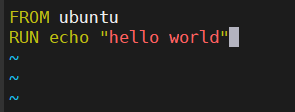
. And start components also be capital letter.



. How to create Dockerfile:-

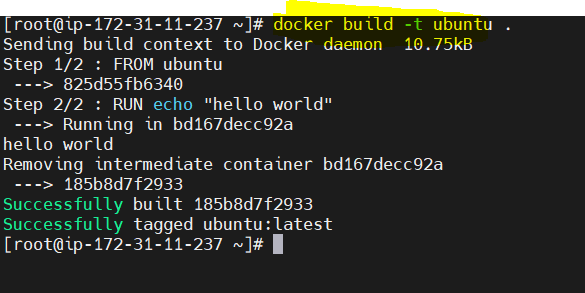
. don’t space on Dockerfile🡺 edit Dockerfile 🡺



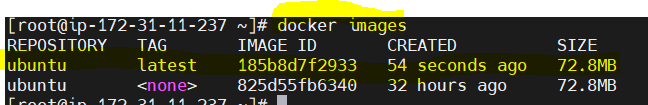


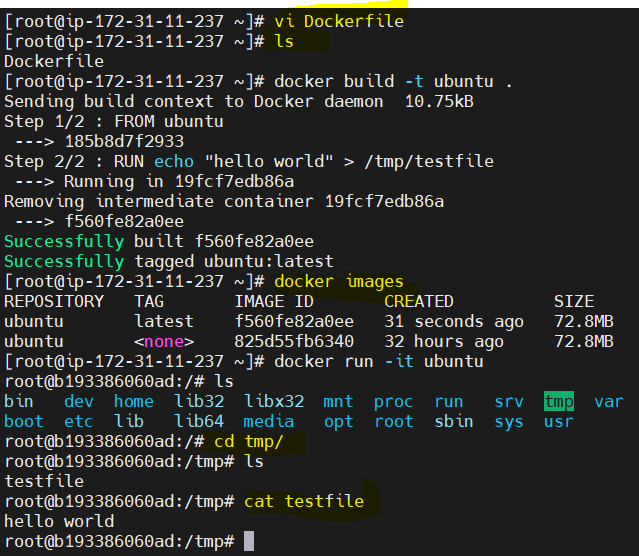
. build file

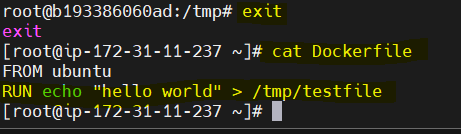
. docker build –t ubuntu . (. Means current directory)



. check the image creation or not







**Docker file components:-**

